LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-215720

Improved Algorithms Speed It Up for Codes

A. Hazi

September 28, 2005

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Improved Algorithms Speed It Up for Codes

HUGE computers, huge codes, complex problems to solve. The longer it takes to run a code, the more it costs. One way to speed things up and save time and money is through hardware improvements—faster processors, different system designs, bigger computers. But another side of supercomputing can reap savings in time and speed: software improvements to make codes—particularly the mathematical algorithms that form them—run faster and more efficiently.

Speed up math? Is that really possible? According to Livermore physicist Eugene Brooks, the answer is a resounding yes.

"Sure, you get great speed-ups by improving hardware," says Brooks, the deputy leader for Computational Physics in N Division, which is part of Livermore's Physics and Advanced Technologies (PAT) Directorate. "But the real bonus comes on the software side, where improvements in software can lead to orders of magnitude improvement in run times."

Brooks knows whereof he speaks. Working with Laboratory physicist Abraham Szöke and others, he has been instrumental in devising ways to shrink the running time of what has, historically, been a tough computational nut to crack: radiation transport codes based on the statistical or Monte Carlo method of calculation (see the box on p. 18). And Brooks is not the only one. Others around the Laboratory, including physicists Andrew Williamson, Randolph Hood, and Jeff Grossman, have come up with innovative ways to speed up Monte Carlo calculations using pure mathematics.

Monte Carlo Not Just for Gamblers

Radiation is energy on the move in the form of light rays or particles such as electrons. Thermal radiation consists of photons, which display characteristics of both high-speed particles and electromagnetic waves. The study of radiation transport deals with predicting and measuring how these photons move through matter. Put simply, thermal radiation transport is a calculational method that examines how heat moves around.

Such calculations are an important part of astrophysical modeling, for example, to simulate stellar evolution, and of inertial confinement fusion modeling. For a mundane example of radiation transport in action, consider a radiant space heater such as those commonly found in homes and garages. Radiant heaters generate invisible infrared radiation that transfers heat not to the air—as convection heaters do—but directly to objects and people themselves. With a radiant space heater, people near the heater begin to feel warmer before the heater has had a chance to raise the temperature of the entire room.

Thermal radiation transport codes have been used at Livermore since the early days of modeling, to simulate how thermally generated photons interact with material. To set up a calculation using these codes, computer scientists divide a material into chunks called zones. Then they incorporate such data as the material's properties and the photons' initial energies, frequencies, and directions of travel. Time is also chopped into discrete steps. Once the problem is set up, the computer grinds through its calculations—step by step, photon by photon, zone by zone—to model how the photons, which transport the thermal energy, move through the material.

Modeling this thermal radiation phenomenon has always been difficult, Brooks notes. For example, a photon's mean free path—the average distance it travels before colliding with another photon—may be shorter than the length of the zone, or its mean free time may be shorter than the time step. These problems are frequently encountered in optically opaque systems such as the interior of stars. Scientists have developed several mathematical methods to solve such problems, including Monte Carlo radiation transport.

Tweaking for Results

Until 1970, the Monte Carlo method used to solve thermal radiation problems was very unstable, Brooks says. "In solving the equations over and over, proceeding through each time step, the numerical solutions had errors that grew over time. It wasn't a physical phenomenon, but a mathematical artifact that popped up in solving the problem on the computer."

In 1971, Joe Fleck and J. D. Cummings worked out an innovative method to dampen this mathematical instability, a scheme they called implicit Monte Carlo (IMC). In essence, they introduced the concept of effective scattering, wherein a fraction of the radiative energy absorbed during a time step is instantly reemitted in all directions before the next time step. In contrast, the more conventional Monte Carlo methods do not emit the absorbed photons until the following step—a process that over time causes the numerical instability. IMC was thus more stable and more accurate than traditional methods for certain situations. However, the effective scattering calculations required a lot of computer time to solve.

Reprinted from Science & Technology Review, May 2004
UCRL-TR-215720

In the late 1980s, Fleck hired Brooks as a postdoctoral fellow to extend the IMC method so that it would be useful for lasers. Brooks developed a technique called symbolic IMC. "This technique removed the scattering problem," he says, "and, instead, it gave us a system of nonlinear mathematical equations, or a matrix, to solve."

Although the symbolic IMC method was faster and cleaner than the original IMC method, its nonlinear system still caused a problem: The noise in opaque materials required large numbers of Monte Carlo particles. Brooks and Szöke returned to this problem in 2003, to try to speed up the calculations. They found that the mathematical noise in the Monte Carlo system corresponds to what happens when a photon is absorbed in the zone in which it originates. This quick absorption of photons happens frequently in opaque materials. "As a result," says Brooks, "when we're modeling an opaque material, we often end up wasting a lot of time using computational power to solve a part of the problem that can easily be done with a pencil and paper."

The breakthrough came when the scientists realized that calculations aren't needed for all of the photons—only for the ones that escape one zone and are transported to the next. Szöke suggested subtracting the calculations of the photons being emitted and reabsorbed—a mathematical construct they called the difference formulation. "So far," Brooks says, "the difference formulation is working very well."

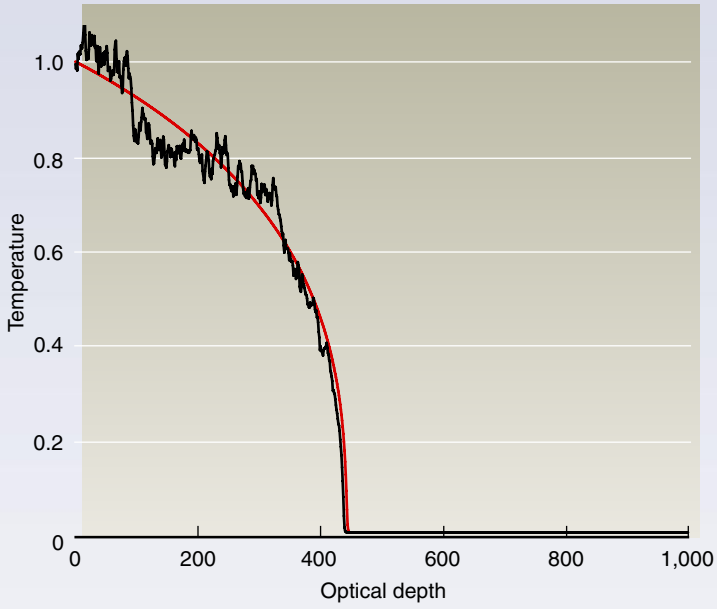
The initial test problem was a one-dimensional simulation of a thick material slab. The simulated slab was divided into many zones with various opacities and time steps. Using the difference formulation increased the algorithm processing speed by factors of up to 1 million, whereas using the older formulation on the massively parallel supercomputers improved speed by factors of 1,000 or so. The makers of supercomputers need not worry, however, because the difference formulation can be adapted to parallel computing, and, says Brooks, the demands of computer users for increased speed are insatiable.

Algorithms in Nanoscience

Mathematical improvements to Monte Carlo methods benefit other Livermore research areas besides astrophysics, including nanosciences. Williamson, Hood, and Grossman, who all work in PAT's Quantum Simulations Group, model systems with only 100 to 200 atoms to better determine their material properties. "At these sizes, quantum mechanical effects can change a material's properties," says Grossman. "For instance, shining laser light at a palm-sized piece of silicon will cause the silicon to emit photons at a wavelength not visible to the human eye. If we shine the same laser light on a silicon quantum dot of 100 atoms (about 2 nanometers square), the dot emits visible light. What's more, the color of the emitted light—whether blue, red, or something in between—will depend on the size of the silicon chunk." (See *S&TR*, November 2003, pp. 4–10.)

Why do materials behave so oddly in such small quantities? The answer can be found in a solution of Schrödinger's equation, which describes the properties of an electron's wave function. In this world of the very small, the electron is treated as a wave, not as a particle. Solving Schrödinger's equation for one particle is simple enough to be done by hand. But as the number of electrons or particles grows, the calculation's complexity grows exponentially, so computers—and lots of computational time—are required to solve the problem.

One approach to these calculations is called the Quantum Monte Carlo (QMC) method. The QMC method uses random numbers to generate an approximate answer with an error bar that indicates the accuracy of the approximation. The smaller the error bar, the more accurate the approximation. To shrink the error bar, the code must choose more random numbers, which increases the program's processing time because the code runs more iterations. Ideally, the



A material's temperature is increased by the transport process of a thermal wave, which is propagating from left to right. The material was initially at 0.01 temperature units. At the start of the simulation, it was abruptly heated on the left side at a temperature of 1 unit. The black curve shows the results from a simulation using the standard formulation. The curve's jagged appearance is caused by the mathematical noise in the calculations. The red curve shows the results from a simulation using the difference formulation. This calculation was performed using the same run time on the same computer as the standard formulation. The Monte Carlo noise in the difference formulation is too small to be shown.

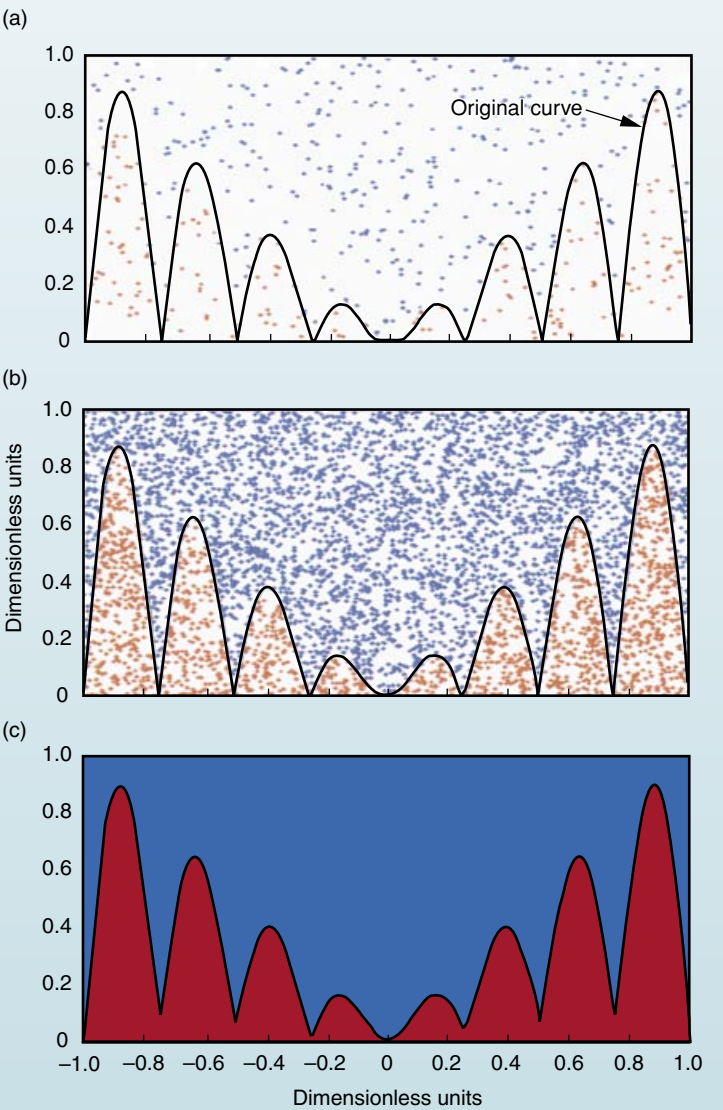
Monte Carlo Primer

In 1946, mathematician Stanislaw Ulam named a set of statistical problem-solving methods "Monte Carlo." The code itself was created at Los Alamos National Laboratory during the Manhattan Project, and the first Monte Carlo calculations were performed in 1948 on the ENIAC, the world's first electronic digital computer. Monte Carlo methods use sequences of random numbers to perform computer simulations. Every simulation is based on events that happen randomly, so the outcome of a calculation is not always absolutely predictable—much like the throw of a dice or turn of a card. Monte Carlo is used routinely in diverse fields, including simulations of radiation transport in Earth's atmosphere and esoteric subnuclear processes in high-energy physics experiments. The difference between Monte Carlo, the method, and Monte Carlo, the gaming capital, is that the method's "game" involves a physical system rather than a game of chance, and its outcome is not a pot of money or stack of chips, but rather the solution to a problem.

Simple Monte Carlo at Work

A simple example of the Monte Carlo method at work is shown in the figure at right, where random numbers are used to calculate the area under a curve as a fraction of the rectangular box encompassing the curve. The original curve is enclosed within a rectangle, and points within the rectangle are chosen at random. The number of points under the curve is then determined as a fraction of the total points chosen. Because the total area of the enclosing rectangle is known, the ratio of the points under the curve to the total points approximates the fraction of the area lying under the curve. As more points are chosen, the approximation becomes more exact.

For example, when only 500 points are chosen, the calculation estimates the area under the curve as 69.6 percent of the total area within the rectangle. But the accuracy of this estimate may be off by as much as 6 percent. Accuracy improves when more random numbers are chosen. With 5,000 points, the area under the curve is estimated to be 63.96 percent, and the error shrinks to 0.29 percent; with 500,000 points, the area is 63.53 percent with an error of 0.13 percent.



A computer code using Monte Carlo calculations can be used to estimate the area under the curve as a fraction of the entire rectangular box. (a) When the code generates 500 random points, the area under the curve is estimated to be 69.6 percent of the rectangle but the error is 5.9 percent. (b) With 5,000 points, the area is estimated to be 63.96 percent, and the error shrinks to 0.29 percent, and (c) with 500,000 points, the area is 63.53 percent with an error of 0.13 percent.

error bar should be smaller than the differences being measured in the calculation. For example, if scientists want to determine whether the light emitted by a particular quantum dot will be blue or red, they set the code to calculate an answer that's accurate enough—that has an error bar small enough—to differentiate between the opposite ends of the visible spectrum.

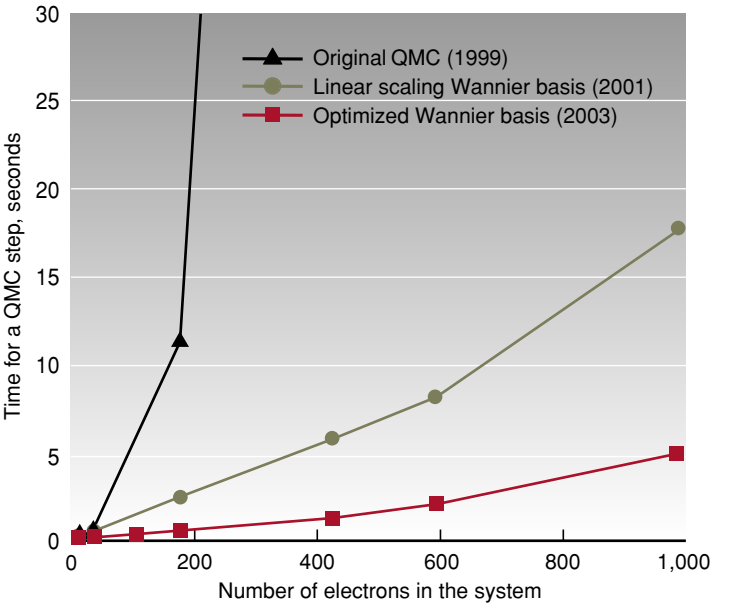
In determining whether an answer can be trusted, scientists can either run the simulation until the error bar is small enough or compare the results with accuracy benchmarks established in physical experiments. "In nanoscience, experiments are difficult to do because of the extremely small scales," says Grossman, "so the ability to use highly accurate benchmark methods such as QMC are quite valuable."

Until recently, however, QMC was only practical when looking at systems composed of small numbers of atoms. "It was a scaling issue," says Grossman. "For instance, if it took 10 minutes for QMC to run a problem with 10 atoms, then running that same problem with 100 atoms required 10,000 minutes—or nearly a week of computational time."

To solve the scaling problem, Grossman, Williamson, and Hood applied a novel mathematical approach called the Wannier basis to the QMC algorithm. Essentially, they performed a mathematical transformation, taking a problem that was difficult to solve and transforming it into a domain where it was easier to solve. For example, one common mathematical transformation is using logarithms, a method for transforming difficult multiplication problems into simpler addition problems. Another is performing a Fourier transform, to transform the wave of a complex electrical signal into simpler sine and cosine waves.

In the original QMC algorithm, the time needed to solve a problem scaled as the cube of the number of atoms involved. Applying this Wannier transformation to QMC produces an algorithm that scales linearly. As a result, the 100-atom system, which previously took a week to process, now requires only 100 minutes.

Williamson is working with Livermore physicist Fernando Reboredo to optimize these Wannier transformations. "We're using nonorthogonal basis functions, which speed up the code another five times," says Williamson. "That increase reduces the run time for the 100-atom system to only 20 minutes. The code also uses eight times less memory, so we can study much larger nanoscience problems."



With a Livermore-developed mathematical transformation called the Wannier basis, the number of electrons in a Quantum Monte Carlo simulation can be increased without a prohibitive increase in the calculation's run time.

Math That Makes a Difference

Even as supercomputing hardware improves, computational scientists, physicists, and others look for better ways to increase the speed of their calculations. Each advance in trimming the time to run a code opens the possibility for simulating a process in more detail and for running multiple simulations in the same amount of time—or even less time—than had been required to process only one.

"Each step forward," says Brooks, "is based on the work that was done before. The advances often happen when people have the opportunity to come together and think differently. It's the collision of people and ideas—through hard work and sudden insights—that leads to these new mathematical constructs, which, in turn, yield faster and in some cases more accurate predictions of phenomenon. In a way, these innovations owe much to serendipity, a lucky roll of the dice—it's Monte Carlo in the scientific realm."

—Ann Parker

Key Words: algorithm, difference formulation, implicit Monte Carlo (IMC) method, Monte Carlo, nanoscience, quantum dots, Quantum Monte Carlo (QMC) method, thermal radiation transport, Wannier basis.

For further information contact Eugene Brooks (925) 423-7341 (eugenebrooks@llnl.gov) or Jeff Grossman (510) 642-8358 (grossman5@llnl.gov).

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.